1. Scalable Deep Learning on Distributed Infrastructures:Challenges, Techniques and Tools (2019)
   1. <https://arxiv.org/pdf/1903.11314.pdf>
   2. One of the driving factors of the success of DL is the scale of training in three dimensions. 1. Model Complexity 2. Training data size 3. Architecture to support large size models with huge data with a short time period.

# Distributed Deep Learning - Part 1 - An Introduction

# <https://joerihermans.com/ramblings/distributed-deep-learning-part-1-an-introduction/>

* 1. <http://cds.cern.ch/record/2276711/files/CERN-THESIS-2018-001.pdf>

1. Parallel and Distributed Deep Learning: A Survey
   1. <https://towardsdatascience.com/parallel-and-distributed-deep-learning-a-survey-97137ff94e4c>
   2. more complex problems that require more and more nodes (even with GPUs).
   3. we communicate between nodes with MPI which is a portable library highly used in supercomputing for inter nodes communication is dominating here.

# Intro to Distributed Deep Learning Systems

* 1. <https://medium.com/@Petuum/intro-to-distributed-deep-learning-systems-a2e45c6b8e7>
  2. How long does it take for the optimization process to reach a [convergence](https://en.wikipedia.org/wiki/Convergence_(logic))? Or, in other words, what is the convergence speed (or rate)?How good is the converged solution?How much training data is needed to guarantee a good solution?
  3. are our models and parameters guaranteed to converge on the same state as without acceleration? How far are we from the true optimal solution?
  4. How much faster (i.e. scalability) we can get if we compare distributed training to non-distributed training? How can we evaluate this?
  5. Data parallelism is quite effective when the size of the **training data is bigger**, since we can scan data much faster with additional nodes. Model parallelism is more applicable to cases in which the size of the **model is too big for one node**,

1. Demystifying Parallel and Distributed Deep Learning: AnIn-Depth Concurrency Analysis
   1. <https://arxiv.org/pdf/1802.09941.pdf>
2. Deep Residual Learning for Image Recognition
   1. <https://arxiv.org/abs/1512.03385>
3. Large Scale Distributed Deep Networks
   1. <https://papers.nips.cc/paper/4687-large-scale-distributed-deep-networks.pdf>
4. SPARKNET: TRAINING DEEP NETWORKS IN SPARK
   1. This is a paper that took local neural network framework and redesigned it to be distributed.
   2. <https://paperswithcode.com/paper/sparknet-training-deep-networks-in-spark>
5. LARGE SCALE DISTRIBUTED NEURAL NETWORK TRAINING THROUGH ONLINE DISTILLATION
   1. <https://arxiv.org/pdf/1804.03235.pdf>
6. Distributed Deep Neural Networks over the Cloud, the Edge and End Devices
   1. <https://paperswithcode.com/paper/distributed-deep-neural-networks-over-the>
7. Massively Distributed SGD: ImageNet/ResNet-50 Training in a Flash
   1. <https://arxiv.org/abs/1811.05233>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-

Somewhat unrelated but this was interesting as it was close to my initial proposal.

It is a natural language process that tries to guess the purpose of the code.

-<https://paperswithcode.com/paper/code2vec-learning-distributed-representations>

Tutorial:

<https://pytorch.org/tutorials/beginner/blitz/data_parallel_tutorial.html#sphx-glr-beginner-blitz-data-parallel-tutorial-py>

<https://pytorch.org/tutorials/beginner/former_torchies/parallelism_tutorial.html>

<https://pytorch.org/tutorials/beginner/blitz/data_parallel_tutorial.html>

<https://pytorch.org/tutorials/intermediate/model_parallel_tutorial.html>

Different Server

<https://pytorch.org/tutorials/intermediate/rpc_tutorial.html>

A presentation

<https://www.uis.no/getfile.php/13458394/Forskning/Vedlegg/10%20IKT/deep_learning_intro.pdf>